Amazon Brazil Market Analysis Using SQL

By Shashwat Singh
NextLeap – Cohort 14
LinkedIn: linkedin.com/in/shashwat-singh-bb2730357 | GitHub: github.com/shashwatanalyst

This document presents SQL-based analysis performed on the Amazon Brazil dataset using PostgreSQL.
Each section includes the business question, SQL logic snapshot, output interpretation, and actionable insights.

All SQL scripts used for this analysis, including schema creation, are provided in the .sql file.

Analysis - I

---

1. **To simplify its financial reports, Amazon India needs to standardize payment values.** Round the average payment values to integer (no decimal) for each payment type and display the results sorted in ascending order.

Query:

```
94
95  -- Objective: Calculate average payment value by payment type
96
97  SELECT
98      payment_type,
99      ROUND(AVG(payment_value),0) AS rounded_avg_payment
100 FROM amazon_brazil.payments
101 WHERE payment_type <> 'not_defined'
102 GROUP BY payment_type
103 ORDER BY rounded_avg_payment ASC;
104
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 4

	payment_type character varying (50)	rounded_avg_payment numeric
1	voucher	66
2	debit_card	143
3	boleto	145
4	credit_card	163

Total rows: 4 | Query complete 00:00:00.136

Insights:

- Voucher-based payments have the lowest average transaction value (66), indicating that vouchers are primarily used for small-ticket purchases.
- Credit cards show the highest average payment value (163), suggesting customers prefer credit cards for higher-value transactions.
- Debit card and boleto payments fall in a similar mid-range, reflecting their use for moderately priced orders.

Recommendations:

- Promote voucher usage for low-value purchases through targeted discounts to increase order frequency.
- Leverage credit-card offers (EMI, cashback) for higher-value orders to maximize revenue.
- Position debit card and boleto options as reliable alternatives for mid-range transactions.

---

2. **To refine its payment strategy, Amazon India wants to know the distribution of orders by payment type.** Calculate the percentage of total orders for each payment type, rounded to one decimal place, and display them in descending order

Query:

```
106
107  -- Objective: Compute percentage distribution of orders by payment type
108
109  SELECT payment_type,
110         ROUND(COUNT(DISTINCT o.order_id)*100.0/
111         (SELECT COUNT(DISTINCT o.order_id)
112          FROM amazon_brazil.payments p
113          JOIN amazon_brazil.orders o
114              ON p.order_id = o.order_id
115          WHERE p.payment_type <> 'not_defined'),1) AS percentage_orders
116  FROM amazon_brazil.payments p
117  JOIN amazon_brazil.orders o
118      ON p.order_id = o.order_id
119  WHERE p.payment_type <> 'not_defined'
120  GROUP BY p.payment_type
121  ORDER BY percentage_orders DESC;
122
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 4

	payment_type character varying (50)	percentage_orders numeric
1	credit_card	76.9
2	boleto	19.9
3	voucher	3.9
4	debit_card	1.5

Total rows: 4 | Query complete 00:00:00.835

Insights:

- Credit cards dominate order payments, accounting for 76.9% of total orders, making them the primary payment method used by customers.
- Boleto contributes a significant share (19.9%), indicating continued reliance on offline or bank-based payment methods.
- Voucher (3.9%) and debit card (1.5%) usage is relatively low, suggesting limited adoption or niche usage.

Recommendations:

- Prioritize optimization and reliability of credit card payment flows, as they support the majority of transactions.
- Maintain boleto as a key payment option, especially for customers preferring non-card payment methods.
- Reassess voucher and debit card offerings to identify opportunities for targeted promotions or user experience improvements.

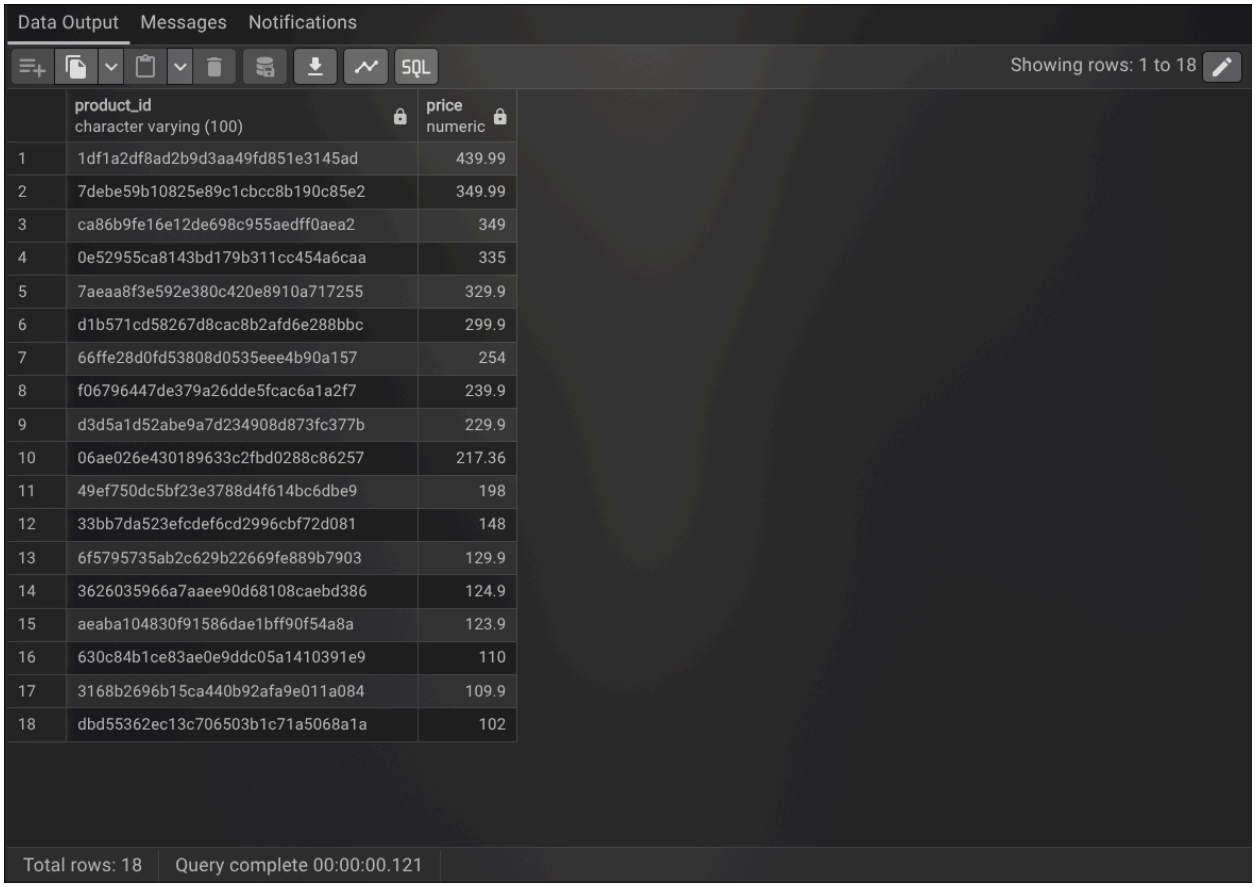
---

3. **Amazon India seeks to create targeted promotions for products within specific price ranges.** Identify all products priced between 100 and 500 BRL that contain the word 'Smart' in their name. Display these products, sorted by price in descending order.

Query:

```
124
125  -- Objective: Identify mid-priced products containing 'Smart' keyword
126
127  SELECT
128      oi.product_id,
129      MAX(oi.price) AS price
130  FROM amazon_brazil.order_items oi
131  JOIN amazon_brazil.products p
132      ON oi.product_id = p.product_id
133  WHERE p.product_category_name ILIKE '%Smart%'
134      AND oi.price BETWEEN 100 AND 500
135  GROUP BY oi.product_id
136  ORDER BY price DESC;
137
```

Output:



	product_id character varying (100)	price numeric
1	1df1a2df8ad2b9d3aa49fd851e3145ad	439.99
2	7debe59b10825e89c1cbcc8b190c85e2	349.99
3	ca86b9fe16e12de698c955aedff0aea2	349
4	0e52955ca8143bd179b311cc454a6caa	335
5	7aeaa8f3e592e380c420e8910a717255	329.9
6	d1b571cd58267d8cac8b2afd6e288bbc	299.9
7	66ffe28d0fd53808d0535eee4b90a157	254
8	f06796447de379a26dde5fcac6a1a2f7	239.9
9	d3d5a1d52abe9a7d234908d873fc377b	229.9
10	06ae026e430189633c2fbd0288c86257	217.36
11	49ef750dc5bf23e3788d4f614bc6dbe9	198
12	33bb7da523efcdef6cd2996cbf72d081	148
13	6f5795735ab2c629b22669fe889b7903	129.9
14	3626035966a7aaee90d68108caebd386	124.9
15	aeaba104830f91586dae1bff90f54a8a	123.9
16	630c84b1ce83ae0e9ddc05a1410391e9	110
17	3168b2696b15ca440b92afa9e011a084	109.9
18	dbd55362ec13c706503b1c71a5068a1a	102

Insights:

- Multiple “Smart” products fall within the 100–500 BRL range, with prices skewed toward the upper end of the band (300–450 BRL), indicating strong presence of mid-to-premium smart products.
- The descending price order highlights a clear pricing ladder, suggesting opportunities for tiered positioning within the “Smart” product segment.
- The volume of qualifying products indicates that “Smart” is a recurring and commercially relevant keyword within this price range.

Recommendations:

- Design tiered promotional campaigns for “Smart” products, differentiating offers for premium-priced (350–500 BRL) versus mid-priced (100–300 BRL) items.
- Highlight “Smart” products prominently in targeted marketing campaigns, as they represent a well-defined and monetizable segment.
- Use price-based bundling or limited-time discounts to improve conversion for higher-priced “Smart” products.

---

4. **To identify seasonal sales patterns, Amazon India needs to focus on the most successful months.** Determine the top 3 months with the highest total sales value, rounded to the nearest integer.

Query:

```
139
140  -- Objective: Identify top 3 months by total sales value
141
142  SELECT
143      TO_CHAR(order_purchase_timestamp,'Month') AS month,
144      ROUND(SUM(oi.price + oi.freight_value),0) AS total_sales
145  FROM amazon_brazil.orders o
146  JOIN amazon_brazil.order_items oi
147      ON oi.order_id = o.order_id
148  WHERE o.order_status NOT IN ('canceled', 'unavailable')
149  GROUP BY month
150  ORDER BY total_sales DESC
151  LIMIT 3;
152
```

Output:

Data Output | Messages | Notifications

SQL | Showing rows: 1 to 3

	month text	total_sales numeric
1	May	1728397
2	August	1658877
3	July	1618537

Total rows: 3 | Query complete 00:00:00.234

Insights:

- May records the highest total sales, indicating a strong peak in customer purchasing activity during late spring.
- August and July follow closely, suggesting sustained high demand during the mid-year period.
- The clustering of top-performing months indicates a seasonal sales concentration rather than isolated spikes.

Recommendations:

- Align major marketing campaigns and promotional events around May to capitalize on peak demand.
- Ensure adequate inventory and logistics capacity during July and August to support sustained sales volume.
- Use historical peak-month trends to improve demand forecasting and seasonal planning.

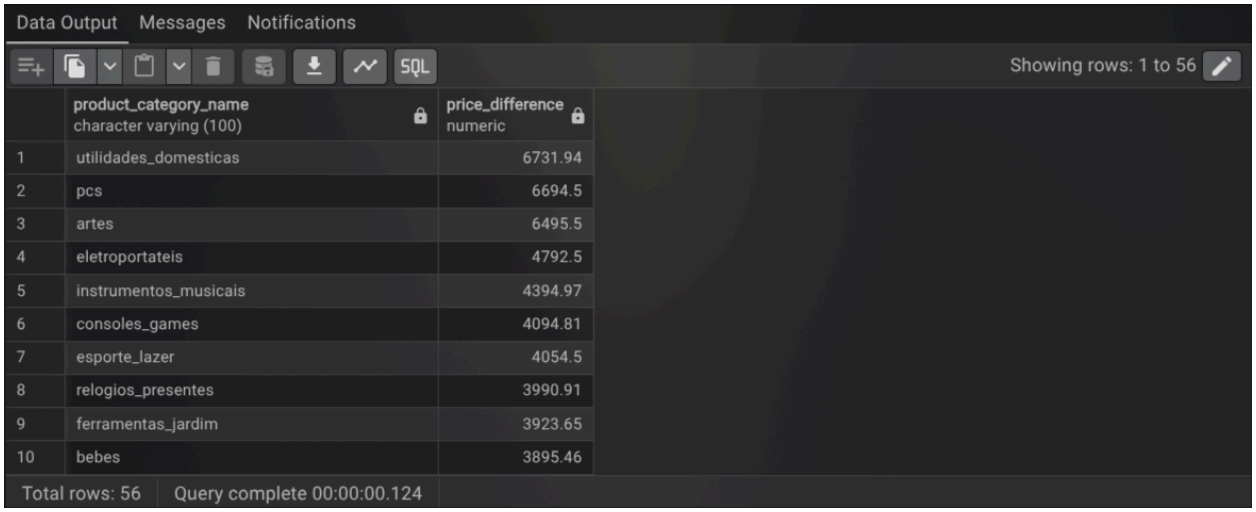
---

5. **Amazon India is interested in product categories with significant price variations.** Find categories where the difference between the maximum and minimum product prices is greater than 500 BRL.

Query:

```
154
155  -- Objective: Find product categories with high price dispersion
156
157  SELECT
158      p.product_category_name,
159      MAX(oi.price)-MIN(oi.price) AS price_difference
160  FROM amazon_brazil.order_items oi
161  JOIN amazon_brazil.products p
162      ON oi.product_id = p.product_id
163  WHERE p.product_category_name IS NOT NULL
164  GROUP BY p.product_category_name
165  HAVING MAX(oi.price) - MIN(oi.price) > 500
166  ORDER BY price_difference DESC;
167
```

Output:



	product_category_name character varying (100)	price_difference numeric
1	utilidades_domesticas	6731.94
2	pcs	6694.5
3	artes	6495.5
4	eletroportateis	4792.5
5	instrumentos_musicais	4394.97
6	consoles_games	4094.81
7	esporte_lazer	4054.5
8	relogios_presentes	3990.91
9	ferramentas_jardim	3923.65
10	bebes	3895.46

Insights:

- Categories such as *utilidades_domesticas, pcs,* and *artes* exhibit very high price dispersion, reflecting a wide mix of low- and high-priced products.
- In contrast, categories such as *audio*, *artigos_de_festas*, and *bebidas* show comparatively lower price variation (though still above 500 BRL), suggesting more standardized pricing structures.
- This contrast highlights that not all high-variation categories require the same pricing or merchandising approach.

Recommendations:

- For categories with very high price variation, introduce clearer price-based segmentation or sub-categories to separate entry-level and premium products.
- For categories with lower price dispersion, maintain consistent pricing strategies and use them for simpler promotional messaging.
- Use price-variation analysis to tailor category-level pricing, filtering, and discount strategies rather than applying a uniform approach across all categories.

---

6. **To enhance the customer experience, Amazon India wants to find which payment types have the most consistent transaction amounts.** Identify the payment types with the least variance in transaction amounts, sorting by the smallest standard deviation first.

Query:

```
169
170  -- Objective: Measure payment type consistency using transaction variance
171
172  SELECT
173      payment_type,
174      ROUND(STDDEV(payment_value),2) AS std_deviation
175  FROM amazon_brazil.payments
176  WHERE payment_type <> 'not_defined'
177  GROUP BY payment_type
178  ORDER BY std_deviation ASC;
179
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 4

	payment_type character varying (50)	std_deviation numeric
1	voucher	115.52
2	boleto	213.58
3	credit_card	222.12
4	debit_card	245.79

Total rows: 4 | Query complete 00:00:00.117

Insights:

- Voucher payments have the lowest standard deviation, indicating the most consistent transaction amounts.
- Boleto and credit card payments show moderate variability, reflecting a wider range of purchase values.
- Boleto and credit card payments show moderate variability, reflecting a wider range of purchase values.

Recommendations:

- Use vouchers for standardized promotions or fixed-value offers to maintain predictable transaction behavior.
- Continue supporting boleto and credit card payments for varied purchase values while monitoring variability.
- Review debit card usage patterns to identify opportunities for improving transaction consistency.

---

7. **Amazon India wants to identify products that may have incomplete name in order to fix it from their end**. Retrieve the list of products where the product category name is missing or contains only a single character.

Query:

```sql
-- Objective: Detect products with missing or incomplete category names

SELECT
    product_id,
    product_category_name
FROM amazon_brazil.products
WHERE product_category_name IS NULL
OR LENGTH(TRIM(product_category_name)) = 1
ORDER BY product_category_name;
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 614

	product_id [PK] character varying (100)	product_category_name character varying (100)
1	ce6f74096c84567f22728c84f3d6e7fc	c
2	c7fce98e1aa3d8a6cbaaebc7ab99cbe6	f
3	afbe1e973aefbf72a330e3bc72d4b476	t
4	3f13f4fabd1eafe564af941a8ee8e279	w
5	5fb61f482620cb672f5e586bb132eae9	[null]
6	e10758160da97891c2fdcbc35f0f031d	[null]
7	39e3b9b12cd0bf8ee681bbc1c130feb5	[null]
8	794de06c32a626a5692ff50e4985d36f	[null]
9	7af3e2da474486a3519b0cba9dea8ad9	[null]
10	629beb8e7317703dcc5f35b5463fd20e	[null]

Total rows: 614 | Query complete 00:00:00.169

Insights:

- A large number of products (600+) have missing or invalid category names, including null values and single-character entries.
- Single-character category values (e.g., “c”, “f”, “t”) indicate data entry or ingestion issues rather than meaningful classifications.
- Incomplete category data can negatively impact reporting accuracy, product discovery, and category-level analysis.

Recommendations:

- Clean and standardize product category names by correcting or removing invalid entries.
- Implement validation rules during data ingestion to prevent null or single-character category values.
- Reprocess historical data after cleanup to ensure accurate category-based reporting and analysis.

Analysis - II

---

1. **Amazon India wants to understand which payment types are most popular across different order value segments (e.g., low, medium, high).** Segment order values into three ranges: orders less than 200 BRL, between 200 and 1000 BRL, and over 1000 BRL. Calculate the count of each payment type within these ranges and display the results in descending order of count

Query:

```sql
-- Objective: Analyze payment preference across order value segments

WITH order_value_table AS (
SELECT order_id ,SUM(price + freight_value) AS order_value
FROM amazon_brazil.order_items
GROUP BY order_id
)

SELECT
    CASE WHEN ov.order_value < 200 THEN 'low'
         WHEN ov.order_value BETWEEN 200 AND 1000 THEN 'medium'
         ELSE 'high'
         END AS order_value_segment,
    p.payment_type,
    COUNT(DISTINCT ov.order_id) AS count
FROM order_value_table ov
JOIN amazon_brazil.payments p
    ON ov.order_id = p.order_id
WHERE p.payment_type <> 'not_defined'
GROUP BY order_value_segment,p.payment_type
ORDER BY count DESC;
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 12

	order_value_segment text	payment_type character varying (50)	count bigint
1	low	credit_card	59585
2	low	boleto	16306
3	medium	credit_card	15456
4	low	voucher	3208
5	medium	boleto	3133
6	low	debit_card	1280
7	high	credit_card	950
8	medium	voucher	531
9	medium	debit_card	227
10	high	boleto	175
11	high	voucher	27
12	high	debit_card	14

Total rows: 12 | Query complete 00:00:00.795

Insights:

- Credit cards are the most popular payment method across all order value segments, dominating low, medium, and high-value orders.
- Low-value orders are heavily concentrated, with significantly higher order volumes compared to medium and high segments.
- Voucher and debit card usage declines sharply as order value increases, indicating limited use for higher-value purchases.

Recommendations:

- Continue prioritizing credit card payment flows, as they support transactions across all order value segments.
- Promote vouchers and debit cards primarily for low-value orders through targeted offers.
- Focus high-value order incentives (e.g., EMI, cashback) on credit cards to maximize conversion.

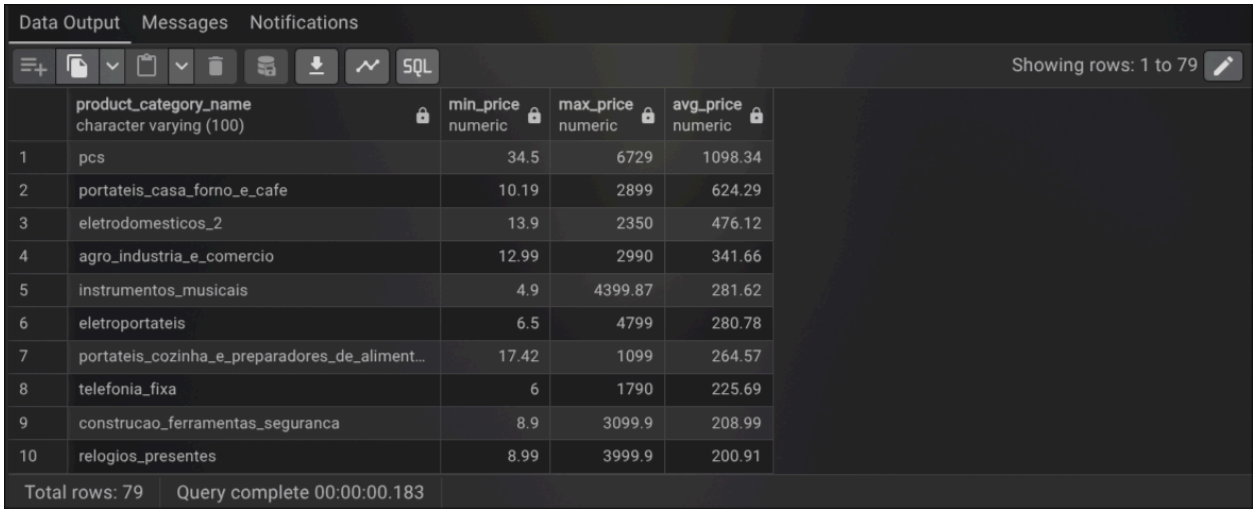
---

2. **Amazon India wants to analyse the price range and average price for each product category.** Calculate the minimum, maximum, and average price for each category, and list them in descending order by the average price.

Query:

```
222
223  -- Objective: Analyze price distribution and average pricing by category
224
225  SELECT
226      p.product_category_name,
227      MIN(oi.price) AS min_price,
228      MAX(oi.price) AS max_price,
229      ROUND(AVG(oi.price),2) AS avg_price
230  FROM amazon_brazil.order_items oi
231  JOIN amazon_brazil.products p
232      ON oi.product_id = p.product_id
233  GROUP BY p.product_category_name
234  ORDER BY avg_price DESC;
235
```

Output:


Data Output Messages Notifications

Showing rows: 1 to 79

	product_category_name character varying (100)	min_price numeric	max_price numeric	avg_price numeric
1	pcs	34.5	6729	1098.34
2	portateis_casa_forno_e_cafe	10.19	2899	624.29
3	eletrodomesticos_2	13.9	2350	476.12
4	agro_industria_e_comercio	12.99	2990	341.66
5	instrumentos_musicais	4.9	4399.87	281.62
6	eletroportateis	6.5	4799	280.78
7	portateis_cozinha_e_preparadores_de_aliment...	17.42	1099	264.57
8	telefonia_fixa	6	1790	225.69
9	construcao_ferramentas_seguranca	8.9	3099.9	208.99
10	relogios_presentes	8.99	3999.9	200.91

Total rows: 79 Query complete 00:00:00.183

Insights:

- Product categories show significant variation in average prices, with *pcs* and *portateis_casa_forno_e_cafe* leading in average price, indicating premium-oriented categories.
- High average-price categories also tend to have wide price ranges, reflecting diverse product tiers within the same category.
- In contrast, categories such as *bebidas*, *alimentos*, and *flores* have consistently lower average prices, indicating standardized, low-ticket offerings.

Recommendations:

- Apply differentiated pricing and promotional strategies for high average-price categories, focusing on premium positioning and value-based messaging.
- Maintain simplified pricing and high-volume strategies for low average-price categories.
- Use average price insights to tailor category-specific marketing and inventory planning.

---

3. **Amazon India wants to identify the customers who have placed multiple orders over time.** Find all customers with more than one order, and display their customer unique IDs along with the total number of orders they have placed.

Query:

```sql
-- Objective: Identify repeat customers based on order frequency

WITH customer_orders AS (
SELECT
    customer_id,
    COUNT(order_id) AS total_orders
FROM amazon_brazil.orders o
GROUP BY customer_id
HAVING COUNT(order_id) > 1
)
SELECT
    c.customer_unique_id,
    co.total_orders
FROM amazon_brazil.customers c
JOIN customer_orders co
    ON c.customer_id = co.customer_id
ORDER BY co.total_orders DESC;

```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 215

	customer_unique_id character varying (100)	total_orders bigint
1	363f980585bf04c1a88fdb986011c52e	16
2	5bf4ea2d98005b960eea0dbf652ef4e7	16
3	75f15790b1852b42b1dbf645d98ffa1c	16
4	c024307523462166b42112cfb6c8e900	16
5	3d364a7768fae99678635c4370295d20	16
6	f9c4e8531c2fe4159beb562fd7c2bd59	16
7	a6168cd79131e64acef92e3c74d6cc43	16
8	f300b00a19af4d4f7bdf9f4524c4587a	16
9	417b909c0962b2610f1cfeb1c1478986	16
10	5f94af52aef02c968a2e0f01f430864e	16

Total rows: 215 | Query complete 00:00:00.415

Insights:

- A large set of customers have placed a very high number of orders (up to 16), indicating strong repeat purchasing behavior.
- Most repeat customers are concentrated at the lower end of the spectrum, with just two orders, suggesting early-stage repeat engagement.
- The wide spread in order counts highlights different levels of customer loyalty within the repeat customer base.

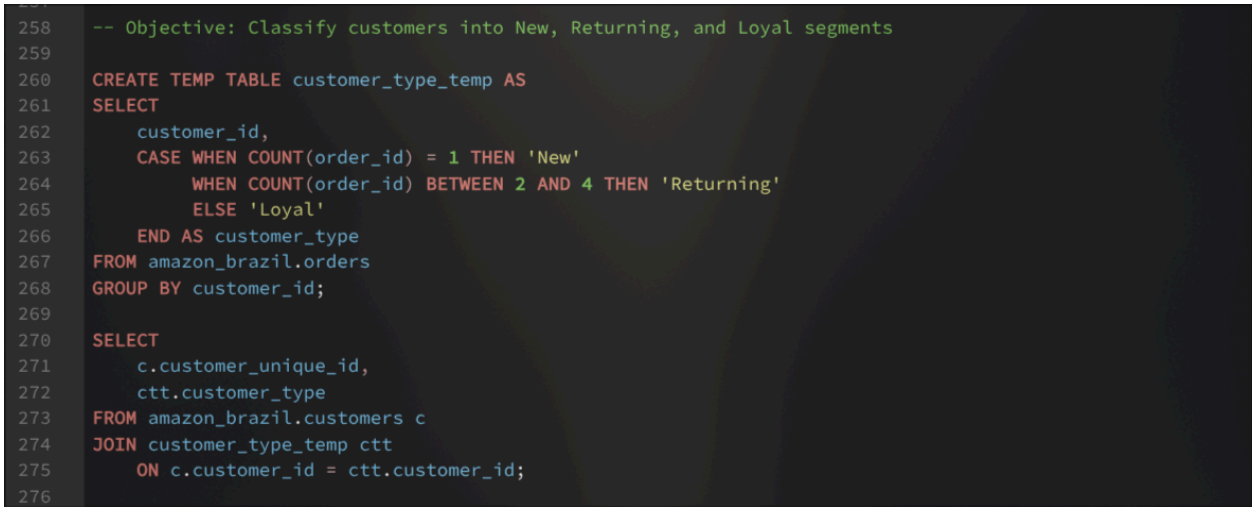
Recommendations:

- Target high-frequency customers with loyalty rewards or exclusive offers to reinforce continued engagement.
- Design re-engagement campaigns for customers with only two orders to encourage progression into higher-frequency segments.
- Use order frequency as a key input for customer segmentation and retention strategies.

---

4. **Amazon India wants to categorize customers into different types ('New – order qty. = 1' ; 'Returning' –order qty. 2 to 4; 'Loyal' – order qty. >4) based on their purchase history.** Use a temporary table to define these categories and join it with the customers table to update and display the customer types.

Query:

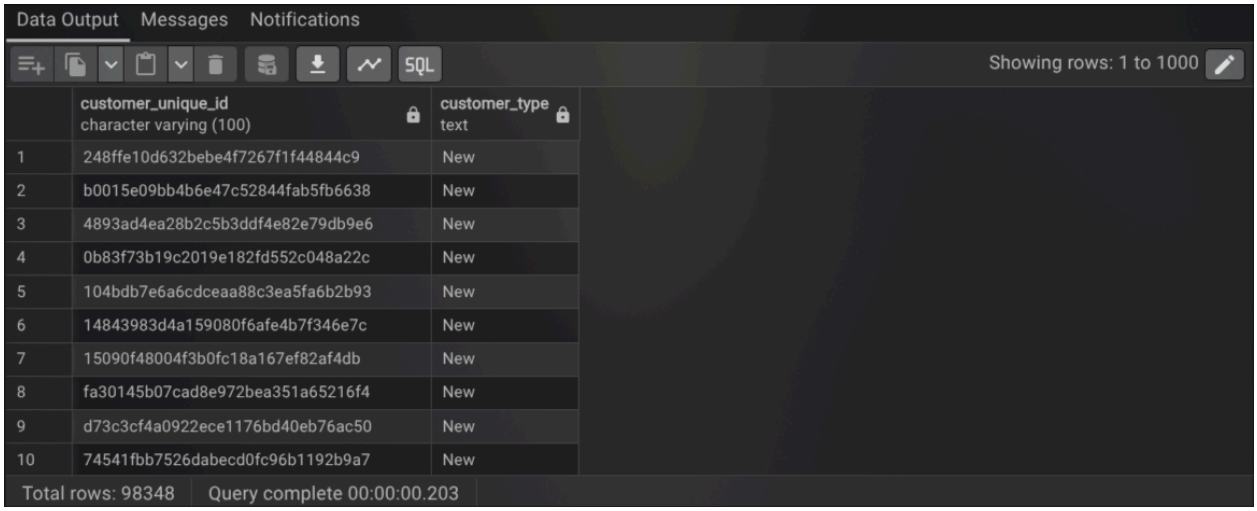


```
-- Objective: Classify customers into New, Returning, and Loyal segments

CREATE TEMP TABLE customer_type_temp AS
SELECT
    customer_id,
    CASE WHEN COUNT(order_id) = 1 THEN 'New'
         WHEN COUNT(order_id) BETWEEN 2 AND 4 THEN 'Returning'
         ELSE 'Loyal'
    END AS customer_type
FROM amazon_brazil.orders
GROUP BY customer_id;

SELECT
    c.customer_unique_id,
    ctt.customer_type
FROM amazon_brazil.customers c
JOIN customer_type_temp ctt
    ON c.customer_id = ctt.customer_id;
```

Output:



Insights:

- The vast majority of customers fall into the “New” category, indicating that most customers have made only a single purchase.
- The dominance of one-time purchasers suggests limited repeat engagement across the customer base.
- Returning and Loyal customers represent a much smaller portion of total customers, highlighting a potential retention gap.

Recommendations:

- Prioritize post-purchase engagement strategies to convert “New” customers into repeat buyers.
- Introduce targeted incentives (e.g., discounts on second purchase) to encourage early repeat behavior.
- Develop loyalty programs focused on increasing order frequency among Returning customers.

---

5. **Amazon India wants to know which product categories generate the most revenue.** Use joins between the tables to calculate the total revenue for each product category. Display the top 5 categories.

Query:

```
-- Objective: Identify repeat customers based on order frequency

WITH customer_orders AS (
SELECT
    customer_id,
    COUNT(order_id) AS total_orders
FROM amazon_brazil.orders o
GROUP BY customer_id
HAVING COUNT(order_id) > 1
)
SELECT
    c.customer_unique_id,
    co.total_orders
FROM amazon_brazil.customers c
JOIN customer_orders co
    ON c.customer_id = co.customer_id
ORDER BY co.total_orders DESC;

```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 215

	customer_unique_id character varying (100)	total_orders bigint
1	363f980585bf04c1a88fdb986011c52e	16
2	5bf4ea2d98005b960eea0dbf652ef4e7	16
3	75f15790b1852b42b1dbf645d98ffa1c	16
4	c024307523462166b42112cfb6c8e900	16
5	3d364a7768fae99678635c4370295d20	16
6	f9c4e8531c2fe4159beb562fd7c2bd59	16
7	a6168cd79131e64acef92e3c74d6cc43	16
8	f300b00a19af4d4f7bdf9f4524c4587a	16
9	417b909c0962b2610f1cfeb1c1478986	16
10	5f94af52aef02c968a2e0f01f430864e	16

Total rows: 215 | Query complete 00:00:00.415

Insights:

- A small set of product categories contributes a significant share of total revenue, with *beleza_saude* leading among all categories.
- The top five categories span both lifestyle and functional products, indicating diversified revenue drivers.
- Revenue concentration in a limited number of categories highlights their strategic importance to overall sales performance.

Recommendations:

- Prioritize inventory planning and marketing investments in top revenue-generating categories to sustain growth.
- Use cross-selling and bundling strategies within these high-performing categories to increase average order value.
- Monitor revenue concentration to identify risks and opportunities for diversification across categories.

Analysis - III

---

1. **The marketing team wants to compare the total sales between different seasons.** Use a subquery to calculate total sales for each season (Spring, Summer, Autumn, Winter) based on order purchase dates, and display the results. Spring is in the months of March, April and May. Summer is from June to August and Autumn is between September and November and rest months are Winter.

Query:

```
300
301  -- Objective: Compare total sales across seasonal periods
302
303  SELECT
304      CASE WHEN DATE_PART('month',order_purchase_timestamp) IN (3,4,5) THEN 'Spring'
305           WHEN DATE_PART('month',order_purchase_timestamp) IN (6,7,8) THEN 'Summer'
306           WHEN DATE_PART('month',order_purchase_timestamp) IN (9,10,11) THEN 'Autumn'
307           ELSE 'Winter'
308           END AS season,
309      ROUND(SUM(oi.price + oi.freight_value),0) AS total_sales
310  FROM amazon_brazil.orders o
311  JOIN amazon_brazil.order_items oi
312      ON o.order_id = oi.order_id
313  WHERE o.order_status NOT IN ('canceled', 'unavailable')
314  GROUP BY season
315  ORDER BY total_sales DESC
316
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 4

	season text	total_sales numeric
1	Spring	4868770
2	Summer	4797448
3	Winter	3363458
4	Autumn	2705851

Total rows: 4 | Query complete 00:00:00.287

Insights:

- Spring records the highest total sales, closely followed by Summer, indicating strong demand during warmer months.
- Winter shows moderate sales performance, while Autumn has the lowest total sales among all seasons.
- Sales exhibit clear seasonality, with higher volumes concentrated in Spring and Summer.

Recommendations:

- Align major marketing campaigns and promotions with Spring and Summer to capitalize on peak demand.
- Use targeted discounts or campaigns during Autumn to stimulate demand in the weakest season.
- Incorporate seasonal trends into sales forecasting and inventory planning.

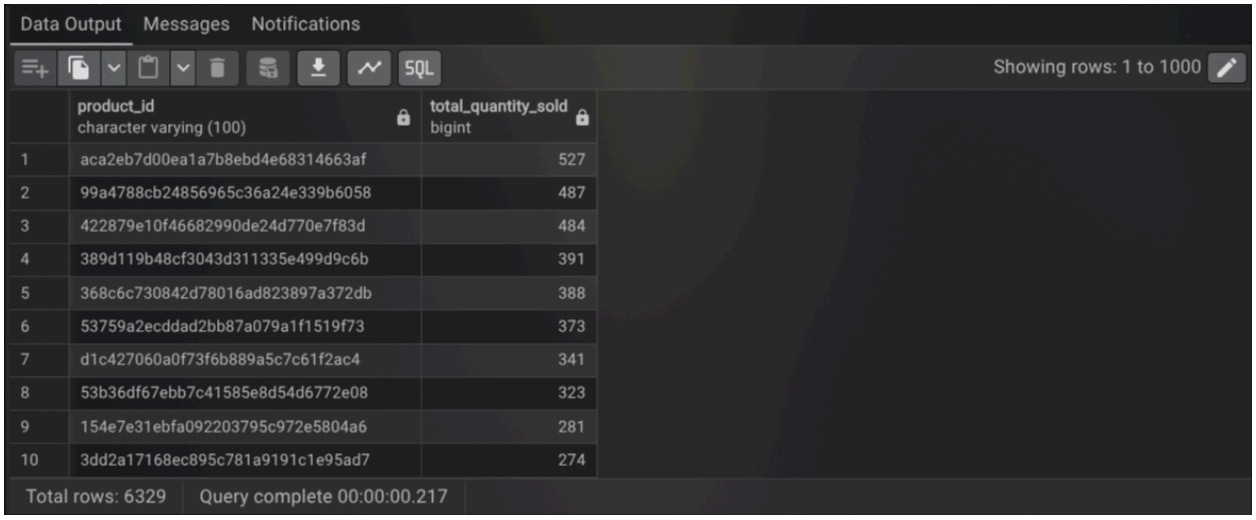
---

2. **The inventory team is interested in identifying products that have sales volumes above the overall average.** Write a query that uses a subquery to filter products with a total quantity sold above the average quantity.

Query:

```
318
319  -- Objective: Identify products with above-average sales volume
320
321  WITH quantity_sold AS (
322  SELECT
323      oi.product_id,
324      COUNT(*) AS total_quantity_sold
325  FROM amazon_brazil.orders o
326  JOIN amazon_brazil.order_items oi
327      ON o.order_id = oi.order_id
328  WHERE o.order_status NOT IN ('canceled', 'unavailable')
329  GROUP BY oi.product_id
330  )
331  SELECT
332      product_id,
333      total_quantity_sold
334  FROM quantity_sold
335  WHERE total_quantity_sold >
336  (
337      SELECT
338          AVG(total_quantity_sold) AS avg_total_quantity_sold
339      FROM quantity_sold
340  )
341  ORDER BY total_quantity_sold DESC;
342
```

Output:



	product_id character varying (100)	total_quantity_sold bigint
1	aca2eb7d00ea1a7b8ebd4e68314663af	527
2	99a4788cb24856965c36a24e339b6058	487
3	422879e10f46682990de24d770e7f83d	484
4	389d119b48cf3043d311335e499d9c6b	391
5	368c6c730842d78016ad823897a372db	388
6	53759a2ecddad2bb87a079a1f1519f73	373
7	d1c427060a0f73f6b889a5c7c61f2ac4	341
8	53b36df67ebb7c41585e8d54d6772e08	323
9	154e7e31ebfa092203795c972e5804a6	281
10	3dd2a17168ec895c781a9191c1e95ad7	274

Insights:

- A subset of products shows significantly higher sales volumes than the overall average, indicating strong and consistent demand.
- Sales volumes among above-average products vary widely, with top products selling several hundred units, while others exceed the average by a smaller margin.
- The long tail of products just above the average highlights a broad base of moderately high-performing items.

Recommendations:

- Prioritize inventory availability and replenishment for top-selling products to avoid stockouts.
- Monitor products slightly above the average to identify emerging high-demand items.
- Use sales volume segmentation to guide inventory planning and demand forecasting.

---

3. **To understand seasonal sales patterns, the finance team is analysing the monthly revenue trends over the past year (year 2018).** Run a query to calculate total revenue generated each month and identify periods of peak and low sales. Export the data to Excel and create a graph to visually represent revenue changes across the months.

Query:

```sql
-- Objective: Analyze monthly revenue trends for year 2018

SELECT
    TO_CHAR(month, 'Mon YYYY') AS month,
    total_revenue
FROM (
    SELECT
        DATE_TRUNC('month', o.order_purchase_timestamp) AS month,
        ROUND(SUM(oi.price + oi.freight_value), 0) AS total_revenue
    FROM amazon_brazil.orders o
    JOIN amazon_brazil.order_items oi
        ON oi.order_id = o.order_id
    WHERE o.order_status NOT IN ('canceled', 'unavailable')
      AND DATE_PART('year', o.order_purchase_timestamp) = 2018
    GROUP BY month
    ORDER BY month
) t
```

Output:

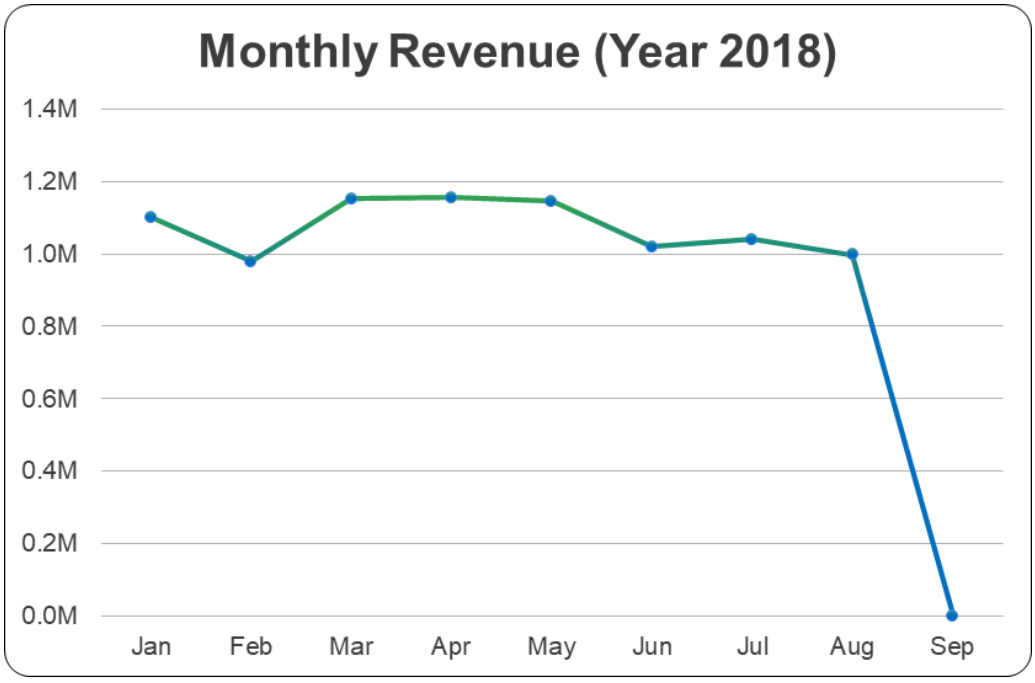
Data Output | Messages | Notifications

Showing rows: 1 to 9

	month text	total_revenue numeric
1	Jan 2018	1101920
2	Feb 2018	979486
3	Mar 2018	1152657
4	Apr 2018	1156249
5	May 2018	1145686
6	Jun 2018	1020382
7	Jul 2018	1039784
8	Aug 2018	996974
9	Sep 2018	166

Total rows: 9 | Query complete 00:00:00.116

Visualization:



Note: September 2018 revenue appears incomplete and may not represent a full month.

Insights:

- Revenue peaks during March to May 2018, indicating a strong sales period in late Q1 and early Q2.
- Revenue remains relatively stable from June to August, with a slight decline compared to the peak months.
- September 2018 shows an unusually low revenue value, suggesting incomplete data for that month rather than an actual drop in sales.

Recommendations:

- Focus marketing and promotional efforts around the March–May period to maximize returns during peak demand.
- Investigate data completeness for September 2018 before using it for trend analysis or forecasting.
- Use sales volume segmentation to guide inventory planning and demand forecasting.

---

4. **A loyalty program is being designed for Amazon India.** Create a segmentation based on purchase frequency: ‘Occasional’ for customers with 1-2 orders, ‘Regular’ for 3-5 orders, and ‘Loyal’ for more than 5 orders. Use a CTE to classify customers and their count and generate a chart in Excel to show the proportion of each segment.

Query:

```
-- Objective: Segment customers based on purchase frequency

WITH CTE AS (
SELECT
    customer_id,
    CASE WHEN COUNT(order_id) <= 2 THEN 'Occasional'
         WHEN COUNT(order_id) BETWEEN 3 AND 5 THEN 'Regular'
         ELSE 'Loyal'
    END AS customer_type
FROM amazon_brazil.orders
GROUP BY customer_id
)
SELECT
    customer_type,
    COUNT(*) AS count
FROM CTE
GROUP BY customer_type
```

Output:

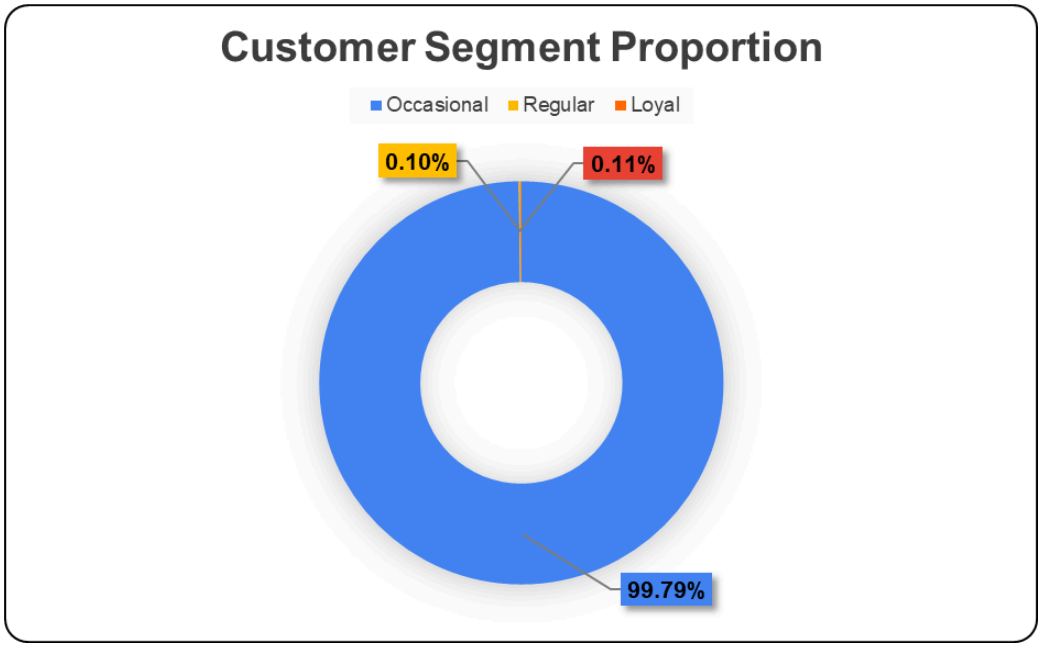
Data Output | Messages | Notifications

Showing rows: 1 to 3

	customer_type text	count bigint
1	Occasional	98144
2	Regular	106
3	Loyal	98

Total rows: 3 | Query complete 00:00:00.311

Visualization:



Note: Regular and Loyal segments together account for less than 0.25% of the customer base.

Insights:

- The customer base is overwhelmingly dominated by Occasional customers, with the vast majority placing only 1–2 orders.
- Regular and Loyal customers together represent a very small fraction of the total customer base.
- This distribution highlights a significant opportunity to improve repeat purchasing behavior.

Recommendations:

- Design the loyalty program to primarily focus on converting Occasional customers into Regular customers through early incentives.
- Offer tiered rewards that encourage progression across segments rather than only rewarding existing Loyal customers.
- Track movement between segments over time to evaluate the effectiveness of the loyalty program.

---

5. **Amazon wants to identify high-value customers to target for an exclusive rewards program.** You are required to rank customers based on their average order value (avg_order_value) to find the top 20 customers.

Query:

```sql
-- Objective: Rank customers by average order value to identify high-value users

SELECT
    o.customer_id ,
    ROUND(AVG(oi.price + oi.freight_value),2) AS avg_order_value,
    RANK()OVER(ORDER BY AVG(oi.price + oi.freight_value) DESC) AS customer_rank
FROM amazon_brazil.order_items oi
JOIN amazon_brazil.orders o
    ON oi.order_id = o.order_id
WHERE o.order_status NOT IN ('canceled', 'unavailable')
GROUP BY o.customer_id
ORDER BY customer_rank
LIMIT 20;
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 20

	customer_id character varying (100)	avg_order_value numeric	customer_rank bigint
1	c6e2731c5b391845f6800c97401a43a9	6929.31	1
2	f48d464a0baaea338cb25f816991ab1f	6922.21	2
3	3fd6777bbce08a352fddd04e4a7cc8f6	6726.66	3
4	df55c14d1476a9a3467f131269c2477f	4950.34	4
5	24bbf5fd2f2e1b359ee7de94defc4a15	4764.34	5
6	3d979689f636322c62418b6346b1c6d2	4681.78	6
7	1afc82cd60e303ef09b4ef9837c9505c	4513.32	7
8	926b6a6fb8b6081e00b335edaf578d35	4194.76	8
9	35a413c7ca3c69756cb75867d6311c0d	4175.26	9
10	e9b0d0eb3015ef1c9ce6cf5b9dcbee9f	4163.51	10
11	3be2c536886b2ea4668eced3a80dd0bb	4042.74	11
12	eb7a157e8da9c488cd4ddc48711f1097	4034.44	12
13	c6695e3b1e48680db36b487419fb0398	4016.91	13
14	31e83c01fce824d0ff786fcd48dad009	3979.55	14
15	addc91fdf9c2b3045497b57fc710e820	3826.80	15
16	19b32919fa1198aefc0773ee2e46e693	3792.59	16
17	66657bf1753d82d0a76f2c4719ab8b85	3736.22	17
18	39d6658037b1b5a07d0a24d423f0bd19	3602.47	18
19	e7c905bf4bb13543e8df947af4f3d9e9	3526.46	19
20	3c7c62e8d38fb18a33a45db8021f2d69	3406.47	20

Total rows: 20 | Query complete 00:00:00.581

Insights:

- The top 20 customers show very high average order values, with the highest exceeding 6,900, indicating a small but extremely valuable customer segment.
- There is a noticeable drop from the top 3 to the rest of the list, suggesting a concentration of value among a few elite customers.
- Even the lowest-ranked customers in the top 20 maintain consistently high average order values, confirming this group's premium nature.

Recommendations:

- Target these top 20 customers with exclusive rewards, personalized offers, or premium service benefits to strengthen retention.
- Introduce VIP tiers or invite-only programs to recognize and incentivize continued high-value purchases.
- Monitor changes in average order value over time to ensure sustained engagement from this segment.

---

6. **Amazon wants to analyze sales growth trends for its key products over their lifecycle.** Calculate monthly cumulative sales for each product from the date of its first sale. Use a recursive CTE to compute the cumulative sales (total_sales) for each product month by month.

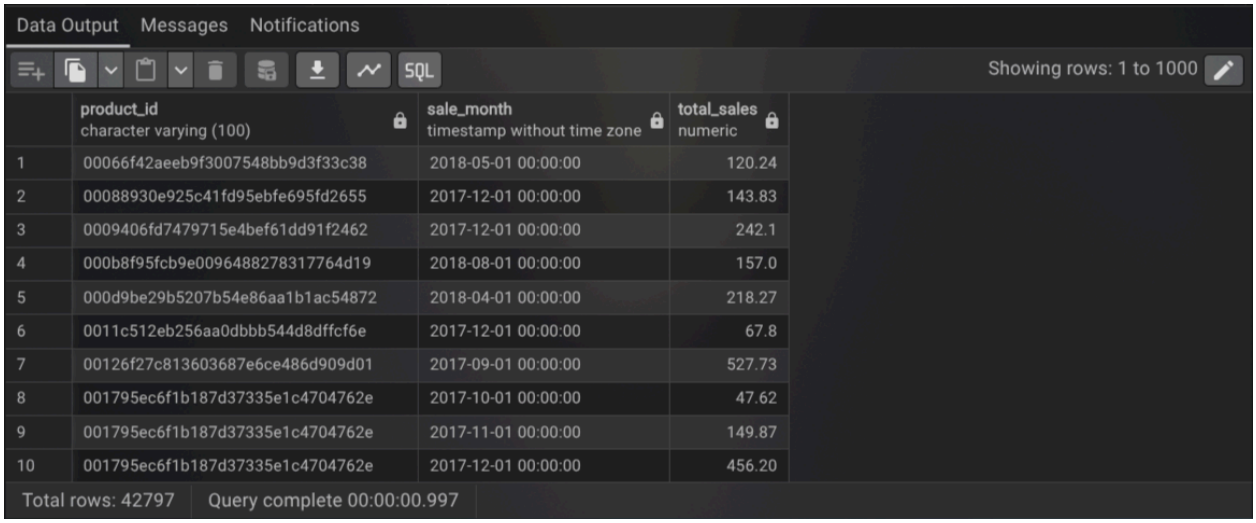
Query:

```
-- Objective: Compute cumulative monthly sales per product over its lifecycle

WITH RECURSIVE monthly_sales AS (
    SELECT
        oi.product_id,
        DATE_TRUNC('month', o.order_purchase_timestamp) AS sale_month,
        SUM(oi.price + oi.freight_value) AS monthly_sales,
        MIN(DATE_TRUNC('month', o.order_purchase_timestamp))
            OVER (PARTITION BY oi.product_id) AS first_sale_month
    FROM amazon_brazil.orders o
    JOIN amazon_brazil.order_items oi
        ON o.order_id = oi.order_id
    WHERE o.order_status NOT IN ('canceled', 'unavailable')
    GROUP BY oi.product_id, sale_month
),
rcte AS (
    -- Anchor: first month per product
    SELECT
        product_id,
        sale_month,
        monthly_sales AS total_sales
    FROM monthly_sales
    WHERE sale_month = first_sale_month

    UNION ALL
    -- Recursive step: move forward month by month
    SELECT
        ms.product_id,
        ms.sale_month,
        rc.total_sales + ms.monthly_sales AS total_sales
    FROM rcte rc
    JOIN monthly_sales ms
        ON ms.product_id = rc.product_id
       AND ms.sale_month = rc.sale_month + INTERVAL '1 month'
)
SELECT *
FROM rcte
ORDER BY product_id, sale_month;
```

Output:



	product_id character varying (100)	sale_month timestamp without time zone	total_sales numeric
1	00066f42aeeb9f3007548bb9d3f33c38	2018-05-01 00:00:00	120.24
2	00088930e925c41fd95ebfe695fd2655	2017-12-01 00:00:00	143.83
3	0009406fd7479715e4bef61dd91f2462	2017-12-01 00:00:00	242.1
4	000b8f95fcb9e0096488278317764d19	2018-08-01 00:00:00	157.0
5	000d9be29b5207b54e86aa1b1ac54872	2018-04-01 00:00:00	218.27
6	0011c512eb256aa0dbbb544d8dffcf6e	2017-12-01 00:00:00	67.8
7	00126f27c813603687e6ce486d909d01	2017-09-01 00:00:00	527.73
8	001795ec6f1b187d37335e1c4704762e	2017-10-01 00:00:00	47.62
9	001795ec6f1b187d37335e1c4704762e	2017-11-01 00:00:00	149.87
10	001795ec6f1b187d37335e1c4704762e	2017-12-01 00:00:00	456.20

Total rows: 42797 Query complete 00:00:00.997

Insights:

- Products exhibit different sales lifecycles, with cumulative sales building gradually from their first sale month rather than showing immediate spikes.
- Some products demonstrate sustained month-over-month growth, while others show limited cumulative growth after initial sales.
- The variation in cumulative sales trajectories indicates differing product performance and lifecycle maturity across the catalog.

Recommendations:

- Identify products with consistently growing cumulative sales and prioritize them for long-term inventory and promotional planning.
- Monitor products with slow cumulative growth to assess whether pricing, visibility, or product relevance needs adjustment.
- Use lifecycle-based cumulative sales trends to inform product launch strategies and discontinuation decisions.

---

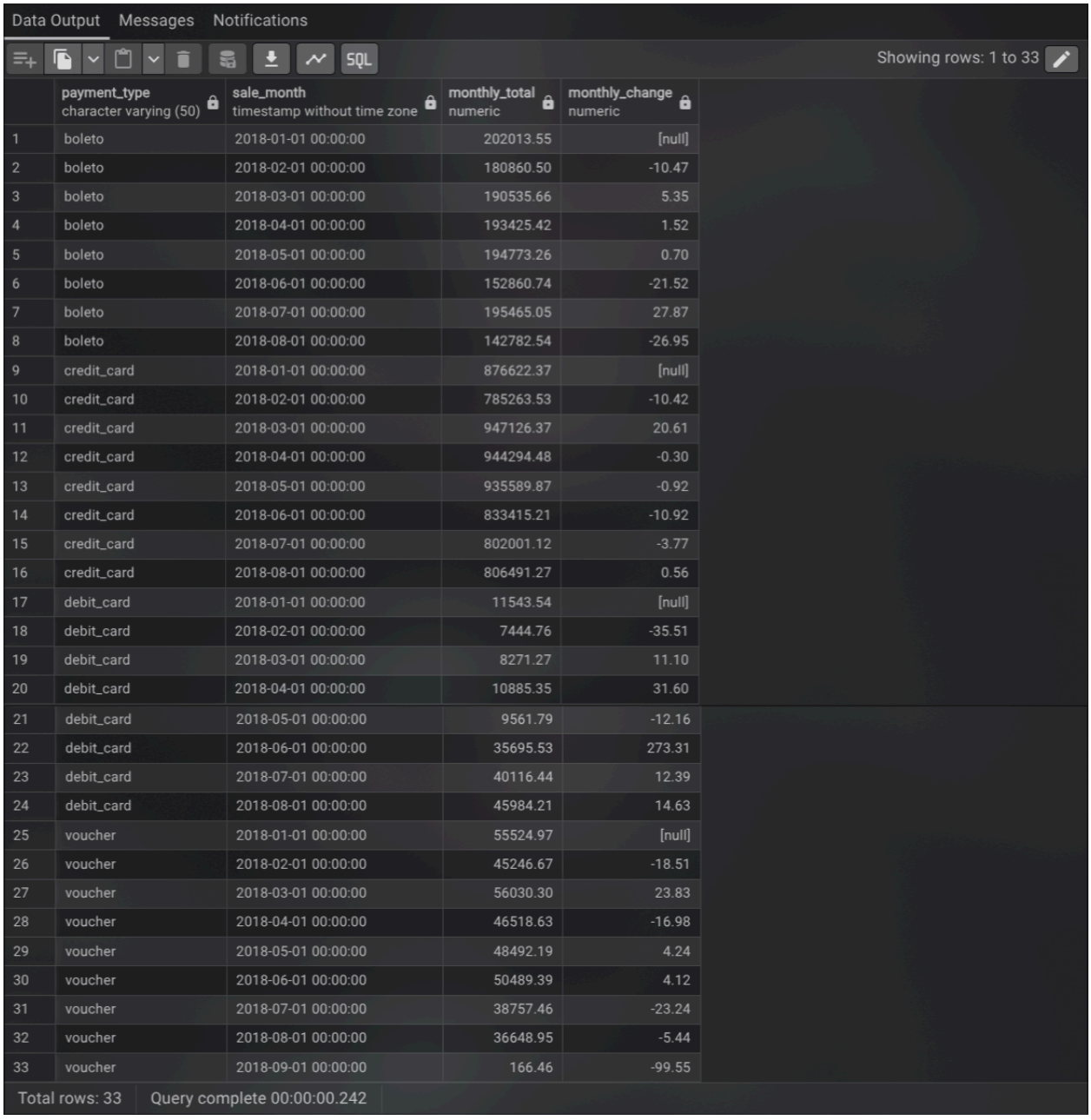
7. **To understand how different payment methods affect monthly sales growth, Amazon wants to compute the total sales for each payment method and calculate the month-over-month growth rate for the past year (year 2018).** Write query to first calculate total monthly sales for each payment method, then compute the percentage change from the previous month.

Query:

```sql
-- Objective: Calculate month-over-month sales growth by payment method

WITH monthly_sales AS (
SELECT
    p.payment_type,
    DATE_TRUNC('month', o.order_purchase_timestamp) AS sale_month,
    SUM(oi.price + oi.freight_value) AS monthly_total
FROM amazon_brazil.orders o
JOIN amazon_brazil.order_items oi
    ON o.order_id = oi.order_id
JOIN amazon_brazil.payments p
    ON p.order_id = o.order_id
WHERE o.order_status NOT IN ('canceled', 'unavailable')
AND DATE_PART('year', o.order_purchase_timestamp) = 2018
GROUP BY p.payment_type, sale_month
ORDER BY p.payment_type, sale_month
)
SELECT payment_type,sale_month,monthly_total,
ROUND((monthly_total - prev_month_total) * 100.0 / prev_month_total,2) AS monthly_change
FROM (
SELECT * ,
LAG(monthly_total)
        OVER(PARTITION BY payment_type) AS prev_month_total
FROM monthly_sales
)t
```

Output:



Data Output | Messages | Notifications

Showing rows: 1 to 33

	payment_type character varying (50)	sale_month timestamp without time zone	monthly_total numeric	monthly_change numeric
1	boleto	2018-01-01 00:00:00	202013.55	[null]
2	boleto	2018-02-01 00:00:00	180860.50	-10.47
3	boleto	2018-03-01 00:00:00	190535.66	5.35
4	boleto	2018-04-01 00:00:00	193425.42	1.52
5	boleto	2018-05-01 00:00:00	194773.26	0.70
6	boleto	2018-06-01 00:00:00	152860.74	-21.52
7	boleto	2018-07-01 00:00:00	195465.05	27.87
8	boleto	2018-08-01 00:00:00	142782.54	-26.95
9	credit_card	2018-01-01 00:00:00	876622.37	[null]
10	credit_card	2018-02-01 00:00:00	785263.53	-10.42
11	credit_card	2018-03-01 00:00:00	947126.37	20.61
12	credit_card	2018-04-01 00:00:00	944294.48	-0.30
13	credit_card	2018-05-01 00:00:00	935589.87	-0.92
14	credit_card	2018-06-01 00:00:00	833415.21	-10.92
15	credit_card	2018-07-01 00:00:00	802001.12	-3.77
16	credit_card	2018-08-01 00:00:00	806491.27	0.56
17	debit_card	2018-01-01 00:00:00	11543.54	[null]
18	debit_card	2018-02-01 00:00:00	7444.76	-35.51
19	debit_card	2018-03-01 00:00:00	8271.27	11.10
20	debit_card	2018-04-01 00:00:00	10885.35	31.60
21	debit_card	2018-05-01 00:00:00	9561.79	-12.16
22	debit_card	2018-06-01 00:00:00	35695.53	273.31
23	debit_card	2018-07-01 00:00:00	40116.44	12.39
24	debit_card	2018-08-01 00:00:00	45984.21	14.63
25	voucher	2018-01-01 00:00:00	55524.97	[null]
26	voucher	2018-02-01 00:00:00	45246.67	-18.51
27	voucher	2018-03-01 00:00:00	56030.30	23.83
28	voucher	2018-04-01 00:00:00	46518.63	-16.98
29	voucher	2018-05-01 00:00:00	48492.19	4.24
30	voucher	2018-06-01 00:00:00	50489.39	4.12
31	voucher	2018-07-01 00:00:00	38757.46	-23.24
32	voucher	2018-08-01 00:00:00	36648.95	-5.44
33	voucher	2018-09-01 00:00:00	166.46	-99.55

Total rows: 33 Query complete 00:00:00.242

Insights:

- Credit card payments generate the highest monthly sales and show relatively stable month-over-month growth.
- Boleto and voucher payments exhibit higher volatility with frequent fluctuations across months.
- Debit card usage shows sharp growth in mid-2018, indicating emerging adoption from a smaller base.
- The sharp drop for voucher payments in September 2018 is likely due to incomplete data rather than an actual decline.

Recommendations:

- Prioritize credit card payment flows as the most stable and reliable revenue driver.
- Align promotions for boleto and voucher payments during months of positive growth to reduce volatility.
- Monitor debit card trends to assess whether recent growth represents sustained behavioral change.
- Exclude incomplete months from growth analysis and forecasting to ensure accuracy.
